

USB SDK 接口说明

版本	日期	说明
V1.0.0	2020/05/27	初版
V1.0.1	2020/10/28	增加新的接口
V1.0.2	2021/02/01	兼容 AT200F/AT21F
V1.0.3	2021/04/30	支持多台连接

目录

简介	4
编译环境.....	4
接口说明.....	5
创建 Handle	5
初始化.....	5
卸载 SDK	5
搜索设备.....	6
打开设备.....	6
关闭设备.....	8
读取 USB 板程序版本.....	8
获取通信类型.....	8
获取机芯类型.....	9
获取测温类型.....	9
通信读指令接口.....	9
通信写指令接口.....	10
快门校正.....	10
色板切换.....	11
色板读取.....	11
读取 SN & PN	12

读取 FPA 温度	12
读取 camera 温度	12
读取面阵宽	13
读取面阵高	13
读取温宽拉伸开关	14
设置温宽拉伸开关	14
设置温宽拉伸低温阈值	14
读取温宽拉伸低温阈值	15
设置温宽拉伸高温阈值	15
读取温宽拉伸高温阈值	15
设置环境变量	16
读取环境变量	16
环境变量生效	17
读取温度单位	17
切换温度单位	17
读取温度解析公式的系数	18
计算边缘检测图像	18
计算边缘增强图像	19

简介

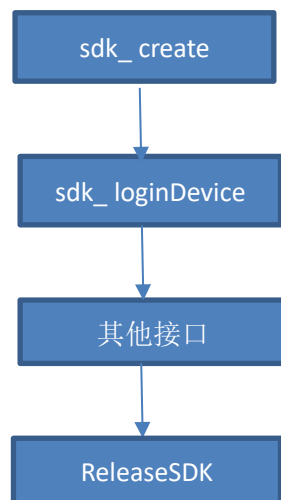
本文档旨在使开发者能快速接入 SDK 进行开发，并能使用 SDK 预览设备图像、测温、串口透传等功能。

编译环境

平台：windows 10 64 位专业版

编译环境：Visual Studio 2015 旗舰版

SDK 调用流程：



接口说明

创建 Handle

函数说明

IRNETHANDLE sdk_create()

创建 handle

参数

无

返回

Handle，用于其他接口传递参数

初始化

函数说明

int sdk_loginDevice(IRNETHANDLE hHandle, HWND hWnd)

初始化 SDK，加载 SDK 资源

参数

[in] hHandle sdk_create 返回值

[in] hWnd 窗口句柄

返回

0 表示成功，-1 表示失败

卸载 SDK

函数说明

void ReleaseSDK(IRNETHANDLE p)

卸载 SDK，释放 SDK 资源

参数

[in] p sdk_create 返回值

返回

无

搜索设备

函数说明

int SearchDevice(IRNETHANDLE p, DeviceLst &devList)

搜索设备

参数

[in] p sdk_create 返回值

DeviceLst 结构体引用

返回

0 表示成功，-1 表示失败

注：DeviceLst 结构体定义如下

```
#define MAX_DEVICE_NUM 50
```

```
struct DeviceInfo
```

```
{  
    int id;          //Device Id  
    char cName[MAX_PATH]; //the Device name  
};
```

```
struct ComName
```

```
{  
    char cComPort[MAX_PATH]; //the Device name  
};
```

```
struct DeviceLst
```

```
{  
    int iComCount; //Number of serial ports  
    int iNumber;   //Device Count  
    DeviceInfo DevInfo[MAX_DEVICE_NUM];  
    ComName ComNameInfo[MAX_DEVICE_NUM];  
};
```

打开设备

函数说明

bool OpenDevice(IRNETHANDLE p, int iGetCurSel, int portIdx)

打开设备，并注册图像回调和温度回调

参数

[in] p sdk_create 返回值
[in] iGetCurSel 设备 ID
[in] portIndx 串口号

返回

True 表示成功，false 表示失败

注：

注册图像回调：

```
void __stdcall SetVideoCallBack(IRNETHANDLE p,VideoCallBack  
pVideoCallBack, void *pContext)
```

参数

[in] p sdk_create 返回值
[in] pVideoCallBack 图像回调函数
[in] pContext 设备上下文

回调函数说明：

```
typedef void(*VideoCallBack)(unsigned char*pBuffer,int  
iWidth,int iHeight, void *pContext);
```

参数

[out] pBuffer 图像数据缓冲区指针
[out] iWidth 面阵宽
[out] iHeight 面阵高
[out] pContext 设备上下文

注册温度回调

```
void __stdcall SetTempCallBack(IRNETHANDLE p,TempCallBack  
pTempCallBack, void *pContext)
```

参数

[in] p sdk_create 返回值
[in] pTempCallBack 温度回调函数
[in] pContext 设备上下文

回调函数说明：

```
typedef void(*TempCallBack)(unsigned char*pBuffer, int
```

`iWidth,int iHeight, void *pContext);`

参数

[out] pBuffer 温度数据缓冲区指针

[out] iWidth 面阵宽

[out] iHeight 面阵高

[out] pContext 设备上下文

注：详细用法请参考 USBSDK_Demo 代码

关闭设备

函数说明

`void CloseDevice(IRNETHANDLE p)`

断开连接，停止图像预览

参数

[in] p sdk_create 返回值

返回

无

读取 USB 板程序版本

函数说明

`DWORD ReadUSBVersion(IRNETHANDLE p)`

读取 USB 板程序版本

参数

[in] p sdk_create 返回值

返回

版本信息

获取通信类型

函数说明

`bool CommunicationType(IRNETHANDLE p)`

获取通信类型

参数

[in] p sdk_create 返回值

返回

True 表示串口通信，false 表示 get/set zoom

获取机芯类型

函数说明

int CoreType(IRNETHANDLE p)

获取机芯类型

参数

[in] p sdk_create 返回值

返回

0 表示失败，1 表示 LT 测温型，2 表示 MicroIII 测温型，3 表示 MicroIII 成像型，4 表示 AT200F，5:AT21F

获取测温类型

函数说明

int TempMeasureType(IRNETHANDLE p)

获取测温类型

参数

[in] p sdk_create 返回值

返回

0 表示失败，1 表示人体测温，2 表示工业测温

通信读指令接口

函数说明

int ReadHandle(IRNETHANDLE p,char* buf, int* pLen)

读取返回指令

参数

[in] p sdk_create 返回值

[out] buf 返回指令缓冲区指针

[out] pLen 返回指令长度

返回

0 表示成功，其他 表示失败

通信写指令接口

函数说明

int WriteHandle(IRNETHANDLE p,char* buf, int len)

发送指令

参数

[in] p sdk_create 返回值
[in] buf 发送指令缓冲区指针
[in] pLen 发送指令长度

返回

0 表示成功，其他 表示失败

注：通信读写指令返回失败，错误代码说明

```
#define OPEN_PORT_FAIL 201 //打开串口失败
#define GET_COM_STATA_FAIL 202 //取串口参数失败
#define SET_COM_STATA_FAIL 203 //设置串口参数失败
#define SET_COM_TIMEOUT_FAIL 204 //设置串口超时失败
#define SEND_DATA_FAIL 205 //发送数据失败
#define RECV_DATA_FAIL 206 //接收数据失败
#define CLOSE_PORT_FAIL 207 //关闭串口失败
#define SEND_DATA_TIMEOUT 208 //发送超时
#define RECV_DATA_TIMEOUT 209 //接收超时
#define ERR_QUERY_INTERFACE_READ_FAIL 210 //get zoom失败
#define ERR_QUERY_INTERFACE_WRITE_FAIL 211 //set zoom失败
```

快门校正

函数说明

int sdk_shutter_correction(IRNETHANDLE p,int iCoreType, int type)

快门校正

参数

[in] p sdk_create 返回值
[in] iCoreType 机芯类型 1 表示 LT 2 或 3 表示 MicroIII 4 表示 AT200F
[in] type 快门校正类型

返回

0 表示成功，-1 表示失败

色板切换

函数说明

`int sdk_set_color_plate(IRNETHANDLE p,int iType, int color_plate)`

色板切换

参数

[in] p sdk_create 返回值

[in] iType 机芯类型 1 表示 LT 2 或 3 表示 Microl III 4 表示 AT200F

[in] color_plate 色板类型

返回

0 表示成功，-1 表示失败

备注：色板编号对应名称依次如下：

WhiteHot;BlackHot;Rainbow;RainbowHC;Iron;Lava;Sky;MidGrey;RdGy;PuOr;Special;Red;IceFire;GreenRed;Special2;RedHot;GreenHot;BlueHot;Green;Blue;

色板读取

函数说明

`int sdk_get_color_plate(IRNETHANDLE p,int iType, int* color_plate)`

色板读取

参数

[in] p sdk_create 返回值

[in] iType 机芯类型 1 表示 LT 其他表示 Microl III

[out] color_plate 色板类型

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

读取 SN & PN

函数说明

```
int sdk_get_SN_PN (IRNETHANDLE p,char *strSN, int *iLenSN, char*  
strPN, int *iLenPN)
```

读取 SN & PN

参数

[in] p sdk_create 返回值
[out] strSN 机芯 SN
[out] iLenSN 机芯 SN 长度
[out] strPN 机芯 PN
[out] iLenPN 机芯 PN 长度

返回

0 表示成功，-1 表示失败

读取 FPA 温度

函数说明

```
int sdk_get_FPA_temp(IRNETHANDLE p,float *fTemp)
```

读取 FPA 温度

参数

[in] p sdk_create 返回值
[out] fTemp FPA 温度

返回

0 表示成功，-1 表示失败

读取 camera 温度

函数说明

```
int sdk_get_camera_temp(IRNETHANDLE p,float *fTemp)
```

读取 camera 温度

参数

[in] p sdk_create 返回值
[out] fTemp camera 温度

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

读取面阵宽

函数说明

`int sdk_get_width(IRNETHANDLE p,int *iValue)`

读取面阵宽

参数

[in] p sdk_create 返回值
[out] iValue 面阵宽

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

读取面阵高

函数说明

`int sdk_get_height(IRNETHANDLE p,int *iValue)`

读取面阵高

参数

[in] p sdk_create 返回值
[out] iValue 面阵高

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

读取温宽拉伸开关

函数说明

```
int sdk_get_wtr_status(IRNETHANDLE p,int* iStatus)
```

读取温宽拉伸开关

参数

[in] p sdk_create 返回值
[out] iStatus 温宽拉伸开关状态 0 表示关 1 表示开

返回

0 表示成功, -1 表示失败

注: AT200F 不支持此接口

设置温宽拉伸开关

函数说明

```
int sdk_set_wtr_status(IRNETHANDLE p,int iStatus)
```

设置温宽拉伸开关

参数

[in] p sdk_create 返回值
[in] iStatus 温宽拉伸开关状态 0 表示关 1 表示开

返回

0 表示成功, -1 表示失败

注: AT200F 不支持此接口

设置温宽拉伸低温阈值

函数说明

```
int sdk_set_wtr_low_threshold(IRNETHANDLE p,int iThreshold)
```

设置温宽拉伸低温阈值

参数

[in] p sdk_create 返回值
[in] iThreshold 温宽拉伸低温阈值*10000

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

读取温宽拉伸低温阈值

函数说明

```
int sdk_get_wtr_low_threshold(IRNETHANDLE p,int* iThreshold)
```

读取温宽拉伸低温阈值

参数

[in] p sdk_create 返回值

[out] iThreshold 温宽拉伸低温阈值*10000

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

设置温宽拉伸高温阈值

函数说明

```
int sdk_set_wtr_high_threshold(IRNETHANDLE p,int iThreshold)
```

设置温宽拉伸高温阈值

参数

[in] p sdk_create 返回值

[in] iThreshold 温宽拉伸高温阈值*10000

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

读取温宽拉伸高温阈值

函数说明

```
int sdk_get_wtr_high_threshold(IRNETHANDLE p,int* iThreshold)
```

读取温宽拉伸高温阈值

参数

[in] p sdk_create 返回值
[out] iThreshold 温宽拉伸高温阈值*10000

返回

0 表示成功，-1 表示失败

注：AT200F 不支持此接口

设置环境变量

函数说明

```
int sdk_set_envir_param(IRNETHANDLE p,envir_param envir_data)
```

设置环境变量

参数

[in] p sdk_create 返回值
[in] envir_data 环境变量结构体

返回

0 表示成功，-1 表示失败

读取环境变量

函数说明

```
int sdk_get_envir_param(IRNETHANDLE p,envir_param* envir_data)
```

读取环境变量

参数

[in] p sdk_create 返回值
[out] envir_data 环境变量结构体

返回

0 表示成功，-1 表示失败

注：环境变量结构体如下

```
typedef struct  
{
```



```

    int emissivity;
    int airTemp;
    int reflectTemp;
    int humidity;
    int distance;
} envir_param; //Parameters are actual values * 10000

```

AT200F 不支持此接口

环境变量生效

函数说明

int sdk_envir_effect(IRNETHANDLE p)

环境变量生效

参数

[in] p sdk_create 返回值

返回

0 表示成功, -1 表示失败

注: AT200F 不支持此接口

读取温度单位

函数说明

int sdk_read_temp_unit(IRNETHANDLE p,unsigned char* ucUnit)

读取温度单位

参数

[in] p sdk_create 返回值

[out] ucUnit 温度单位 0 表示摄氏度 1 表示开尔文 2 表示华氏度

返回

0 表示成功, -1 表示失败

注: AT200F 不支持此接口

切换温度单位

函数说明

`int sdk_set_temp_unit(IRNETHANDLE p,unsigned char ucUnit)`

切换温度单位

参数

[in] p sdk_create 返回值

[in] ucUnit 温度单位 0 表示摄氏度 1 表示开尔文 2 表示华氏度

返回

0 表示成功, -1 表示失败

注: AT200F 不支持此接口

读取温度解析公式的系数

函数说明

`int sdk_get_temp_coefficient(IRNETHANDLE p, int gain, short* param1, short* param2)`

读取温度解析公式的系数

参数

[in] p sdk_create 返回值

[in] gain 增益 0 高增益 1 低增益

[out] param1 系数 1

[out] param2 系数 2

返回

0 表示成功, -1 表示失败

具体使用方法参考 demo 代码

计算边缘检测图像

函数说明

`int sdk_edge_detect(IRNETHANDLE p, unsigned char* imageSrc, unsigned char* imageDst, int width, int height, int level)`

计算边缘检测图像

参数

[in] p sdk_create 返回值
[in] imageSrc 原始图像数据的灰度数据，调用方负责释放
[out] imageDst 经过计算的图像数据，调用方负责释放
[in] width 图像分辨率宽
[in] height 图像分辨率高
[in] level 边缘图像强度，范围 0~100，数值越大，边缘检测信息越稀疏

返回

0 表示成功，-1 表示失败

计算边缘增强图像

函数说明

```
int sdk_edge_detect(IRNETHANDLE p, unsigned char* imageSrc,  
unsigned char* imageDst, int width, int height, int level)
```

计算边缘增强图像

参数

[in] p sdk_create 返回值
[in] imageSrc 原始图像数据的灰度数据，调用方负责释放
[out] imageDst 经过计算的图像数据，调用方负责释放
[in] width 图像分辨率宽
[in] height 图像分辨率高
[in] level 边缘图像强度，范围 0~100，数值越大，边缘信息越强，图像过曝

返回

0 表示成功，-1 表示失败

边缘检测和边缘增强测试样例

```
int main() {  
    cv::Mat src = cv::imread("640.bmp", 0);  
    if (src.empty()) {  
        return -1;  
    }  
    int width = src.cols;  
    int height = src.rows;  
    uchar* srcT = (uchar*)malloc(width*height);  
    uchar* dstT = (uchar*)malloc(width*height);  
    memcpy(srcT, src.data, width* height);  
    fucEdgeEnhance(srcT, dstT, width, height, 10);  
    cv::Mat usm = src.clone();
```

```
memcpy(usm.data, dstT, width* height);  
imshow("mask image", usm);  
cv::imshow("src", src);  
cv::waitKey(0);  
free(srcT);  
free(dstT);  
return 0;  
}
```